AMENDMENTS TO THE CLAIMS

1. (Previously Presented) A method of providing monotonic sequence number, comprising the steps of:

(a) establishing a primary sequence number generator;

(b) establishing a secondary sequence number generator;

(c) generating a sequence number request at an originating node;

(d) forwarding the sequence number request to the primary sequence number generator;

(e) forwarding a sequence number response to the secondary sequence number generator;

(f) storing the sequence number response at the secondary sequence number generator; and

(g) forwarding the response sequence number from the secondary sequence number generator to the originating node.

2. (Original) The method of claim 1, wherein step (c) comprises the sub-steps of:

performing a sequence number request call by a software process operating upon a host processor system to a user-space function; and

executing a sequence number request via an interface located at the originating node.

3. (Original) The method of claim 2 further comprising the step of:

receiving the sequence number response at the interface; and

communicating the sequence number response to the software process.

4. (Original) The method of claim 3, wherein the software process executes a spin loop while waiting to receive the sequence number response.

5. (Original) The method of claim 4, wherein the software process is placed into a sleep state if the sequence number response is not received within a predetermined amount of time.

6. (Original) The method of claim 2, wherein the interface is a hardware card linked to a sequence number fabric.

7. (Original) The method of claim 6, wherein the host processor system comprises a second hardware card linked to a duplicate sequence number fabric.

8. (Previously Presented) The method of claim 2, wherein the primary and secondary sequence number generators store current sequence numbers in memory associated with respective host processor systems via direct memory access operations.

9. (Previously Presented) The method of claim 6, wherein the hardware card pipelines a plurality of sequence number requests.

10. (Currently Amended) A sequence number generation system for providing monotonic sequence numbers, comprising:

a plurality of sequence number devices, connected via a fabric, including at least a primary sequence number generator and a secondary sequence number generator;

the primary sequence number generator disposed to [[a]] receive a sequence number request from an originating device and to a forward sequence number response to the secondary sequence number generator; and

the secondary sequence number generator disposed to receive the sequence number response, store the sequence number response in memory, and forward the response to the originating device.

11. (Previously Presented) The system of claim 10, wherein each sequence number device includes lower level sequence number routines accessible by software processes operating on a host processor system via user-space functions.

12. (Original) The system of claim 11, wherein the user-space functions includes a request new sequence number function.

13. (Original) The system of claim 12, wherein the request new sequence number function causes a requesting application process to execute a spin loop while waiting for receipt of a new sequence number.

14. (Original) The system of claim 13, wherein the requesting application process is placed in a sleep state if the new sequence number is not received within a predetermined amount of time.

15. (Original) The system of claim 10, wherein the primary and secondary sequence number generators store current sequence numbers in memory associated with respective host processor systems via direct memory access operations.

16. (Original) The system of claim 10, wherein the sequence number request is associated with a pipeline of sequence number requests form the originating device.

17-20. (Canceled)

21. (Previously Presented) A distributed computing system, comprising:

a plurality of nodes that each comprise at least one processor and at least one sequence number device;

a fabric interconnecting said sequence number devices of said plurality of nodes;

wherein a respective application is executed on a processor of each of said plurality of nodes that uses sequence numbers issued monotonically within said plurality of nodes and said sequence number devices of said plurality of nodes comprises a primary sequence number generator and a secondary sequence number generator;

wherein each of said applications performs a function call to obtain a sequence number, said function call invoking a sequence number routine of a sequence number device that communicates a sequence number request to said primary sequence generator, said primary sequence number generator monotonically generating a sequence number in response to said request and communicating said generated sequence number to said secondary sequence number generator, said secondary sequence number generator storing said generated sequence number and forwarding said generated sequence number to said originating sequence number device, said originating sequence number device returning said generated sequence number in response to said function call.

22. (Previously Presented) The distributed computing system of claim 21 wherein said originating sequence number device performs a direct memory access to return said generated sequence number.

23. (Previously Presented) The distributed computing system of claim 21 wherein a processor executing an application that has performed said function call is placed in a spin loop while waiting for said generated sequence number to be returned.

24. (Previously Presented) The distributed computing system of claim 21 wherein an application performing said function call is placed in a sleep state when said generated sequence number is not received within a predetermined period of time.

25. (Previously Presented) The distributed computing system of claim 21 wherein a management process, executed on a processor of said plurality of nodes, configures said sequence number devices to select said primary and secondary sequence number generators.

26. (Previously Presented) The distributed computing system of claim 25 wherein, when said management process detects a failure of said primary sequence number generator, said management process selects said secondary sequence number generator to serve as a new primary sequence number generator.

27. (Previously Presented) The distributed computing system of claim 25 wherein said management process communicates an identity of said new primary sequence number generator to said sequence number devices of said plurality of nodes.